Logic Beyond Formulas: Designing a Proof System for Logical Time

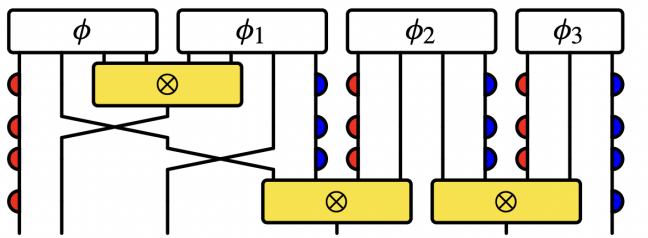
Matteo Acclavio

ROMA TRE
UNIVERSITÀ DEGLI STUDI

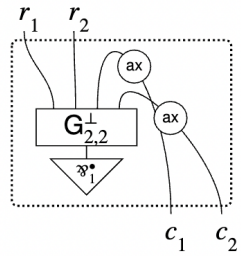
Based in on joint works with Lutz Straßburger, Ross Horne and Sjouke Mauw

XXVII Incontro AILA 13/09/2022

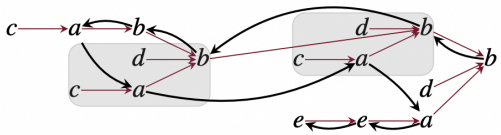
Why I changed the part of the title “Designing Graphical Proof Systems”



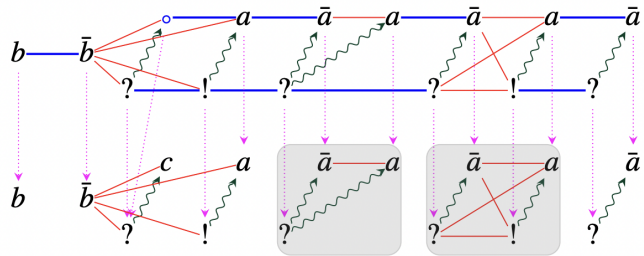
String Diagrams



Generalized Proof Nets



Game Semantics



Combinatorial Proofs

Logic Beyond Formulas: Designing a Proof System for Logical Time

Matteo Acclavio

ROMA TRE
UNIVERSITÀ DEGLI STUDI

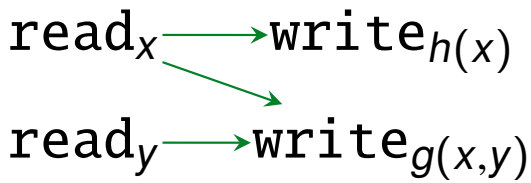
Based in on joint works with Lutz Straßburger, Ross Horne and Sjouke Mauw

XXVII Incontro AILA 13/09/2022

1. Logical Time
2. Why *Graphs*?
3. Preliminaries on Graphs
   - Modular Decomposition
   - Graphical Connectives
4. Graphical proof systems
   - On Deep Inference
   - The rules
   - Transitivity of $\multimap$
   - Conservativity
5. Conclusions and Future Works

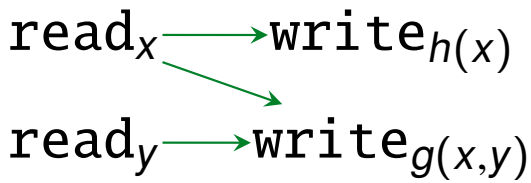
Logical Time

Happens-before relation is crucial in distributed systems



[1] Lamport '78

Happens-before relation is crucial in distributed systems

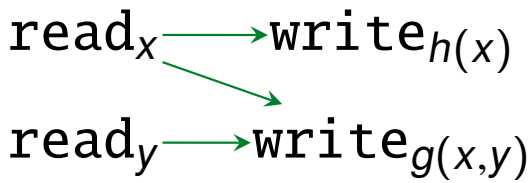


Logical Clocks[1] = Enforcing specific total orders on events

✓	$\mathtt{read}_x \triangleleft \mathtt{write}_{h(x)} \triangleleft \mathtt{read}_y \triangleleft \mathtt{write}_{g(x,y)}$
✓	$\mathtt{read}_x \triangleleft \mathtt{read}_y \triangleleft \mathtt{write}_{g(x,y)} \triangleleft \mathtt{write}_{h(x)}$
✓	$\mathtt{read}_x \triangleleft \mathtt{read}_y \triangleleft \mathtt{write}_{h(x)} \triangleleft \mathtt{write}_{g(x,y)}$
✓	$\mathtt{read}_y \triangleleft \mathtt{read}_x \triangleleft \mathtt{write}_{g(x,y)} \triangleleft \mathtt{write}_{h(x)}$
✓	$\mathtt{read}_y \triangleleft \mathtt{read}_x \triangleleft \mathtt{write}_{h(x)} \triangleleft \mathtt{write}_{g(x,y)}$
✗	$\mathtt{read}_x \triangleleft \mathtt{write}_{h(x)} \triangleleft \mathtt{write}_{g(x,y)} \triangleleft \mathtt{read}_y$

[1] Lamport '78

Happens-before relation is crucial in distributed systems



Logical Clocks[1] = Enforcing specific total orders on events

✓	$\mathtt{read}_x \triangleleft \mathtt{write}_{h(x)} \triangleleft \mathtt{read}_y \triangleleft \mathtt{write}_{g(x,y)}$
✓	$\mathtt{read}_x \triangleleft \mathtt{read}_y \triangleleft \mathtt{write}_{g(x,y)} \triangleleft \mathtt{write}_{h(x)}$
✓	$\mathtt{read}_x \triangleleft \mathtt{read}_y \triangleleft \mathtt{write}_{h(x)} \triangleleft \mathtt{write}_{g(x,y)}$
✓	$\mathtt{read}_y \triangleleft \mathtt{read}_x \triangleleft \mathtt{write}_{g(x,y)} \triangleleft \mathtt{write}_{h(x)}$
✓	$\mathtt{read}_y \triangleleft \mathtt{read}_x \triangleleft \mathtt{write}_{h(x)} \triangleleft \mathtt{write}_{g(x,y)}$
✗	$\mathtt{read}_x \triangleleft \mathtt{write}_{h(x)} \triangleleft \mathtt{write}_{g(x,y)} \triangleleft \mathtt{read}_y$

Logical Time = Happens-before relation without clocks

[1] Lamport '78

Aim of this line of works:

Proof Theory treating the happens-before relation “logically”

Aim of this line of works:

Proof Theory treating the happens-before relation “logically”

That is:

Logical time is expressed by logical connectives

A “happens before” B $\quad\rightsquigarrow\quad$ $A \triangleleft B$

Aim of this line of works:

Proof Theory treating the happens-before relation “logically”

That is:

Logical time is expressed by logical connectives

A “happens before” B $\quad\rightsquigarrow\quad$ $A \triangleleft B$

and

Logical implication ($\multimap$) capturing partial order refinements

Why *Graphs*?

Previous attempts: Pomset logic

Pomset formulas

$$A, B ::= a \mid a^{\perp} \mid A \parr B \mid A \triangleleft B \mid A \otimes B$$

$\parr$	$\triangleleft$	$\otimes$
disjunction	happens-before	conjunction
parallelism	sequentiality	“independence”

Previous attempts: Pomset logic

Pomset formulas

$$A, B ::= a \mid a^{\perp} \mid A \parr B \mid A \triangleleft B \mid A \otimes B$$

$\parr$	$\triangleleft$	$\otimes$
disjunction	happens-before	conjunction
parallelism	sequentiality	"independence"
(commutative)	(non-commutative)	(commutative)

Previous attempts: Pomset logic

Pomset formulas

$$A, B ::= a \mid a^{\perp} \mid A \parr B \mid A \triangleleft B \mid A \otimes B$$

$\parr$	$\triangleleft$	$\otimes$
disjunction	happens-before	conjunction
parallelism (commutative)	sequentiality (non-commutative)	"independence" (commutative)

Negation $(\cdot)^{\perp}$ such that:

$$A^{\perp\perp} = A$$

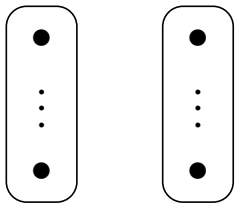
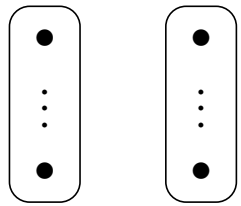
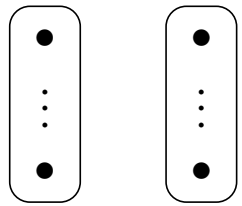
$$(A \parr B)^{\perp} = A^{\perp} \otimes B^{\perp} \qquad (A \triangleleft B)^{\perp} = A^{\perp} \triangleleft B^{\perp} \qquad (A \otimes B)^{\perp} = A^{\perp} \parr B^{\perp}$$

Implication defined "classically":

$$A \multimap B := A^{\perp} \parr B$$

Formulas and Graphs

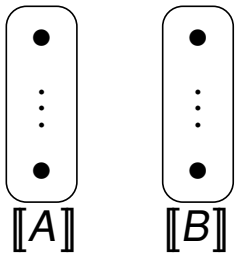
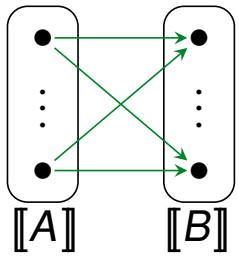
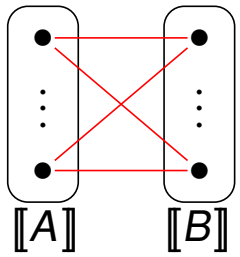
Relation webs[2] = graphs encoding Pomset formulas

$[\![a]\!]$	$[\![a^\perp]\!]$	$[\![A \parr B]\!] = [\![A]\!] \parr [\![B]\!]$	$[\![A \triangleleft B]\!] = [\![A]\!] \triangleleft [\![B]\!]$	$[\![A \otimes B]\!] = [\![A]\!] \otimes [\![B]\!]$
a	$a^\perp$			

[2] Retoré '93, Guglielmi '07

Formulas and Graphs

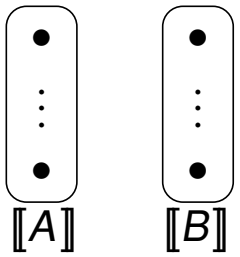
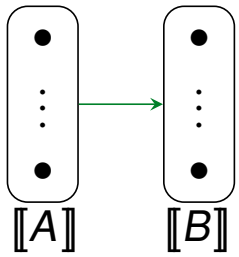
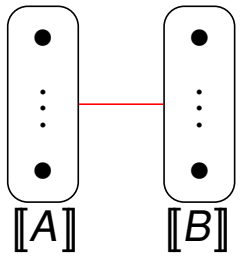
Relation webs[2] = graphs encoding Pomset formulas

$[\![a]\!]$	$[\![a^\perp]\!]$	$[\![A \parr B]\!] = [\![A]\!] \parr [\![B]\!]$	$[\![A \triangleleft B]\!] = [\![A]\!] \triangleleft [\![B]\!]$	$[\![A \otimes B]\!] = [\![A]\!] \otimes [\![B]\!]$
a	$a^\perp$			

[2] Retoré '93, Guglielmi '07

Formulas and Graphs

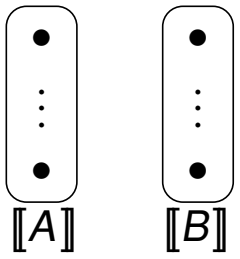
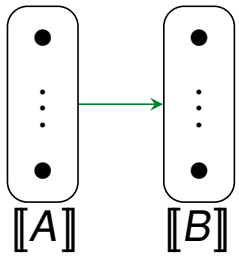
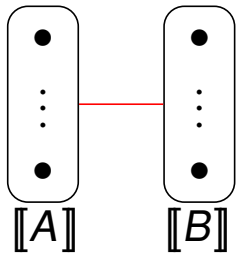
Relation webs[2] = graphs encoding Pomset formulas

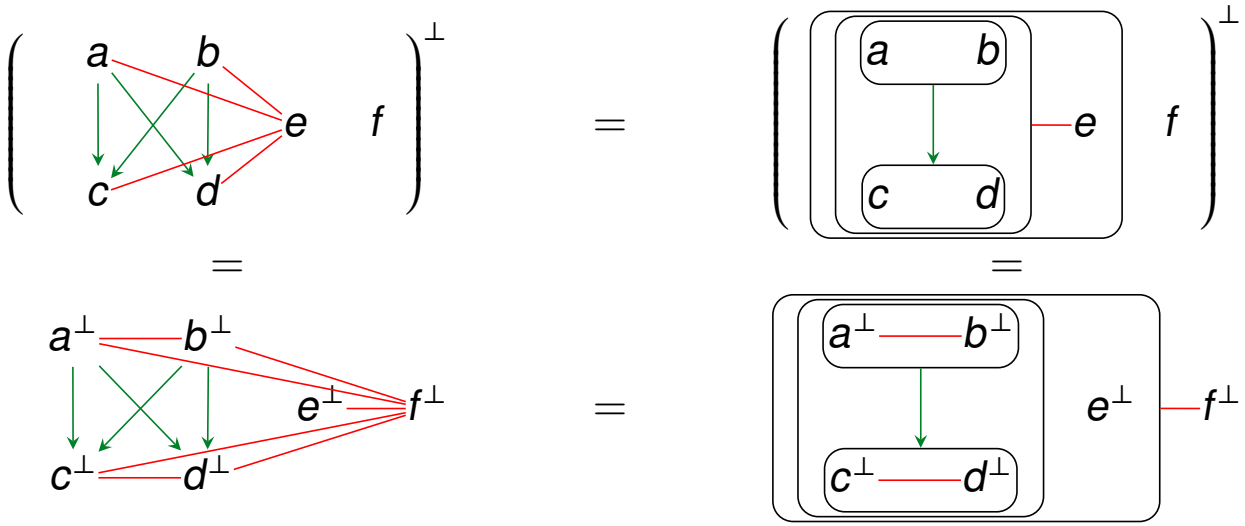
$[\![a]\!]$	$[\![a^\perp]\!]$	$[\![A \parr B]\!] = [\![A]\!] \parr [\![B]\!]$	$[\![A \triangleleft B]\!] = [\![A]\!] \triangleleft [\![B]\!]$	$[\![A \otimes B]\!] = [\![A]\!] \otimes [\![B]\!]$
a	$a^\perp$			

[2] Retoré '93, Guglielmi '07

Formulas and Graphs

Relation webs[2] = graphs encoding Pomset formulas

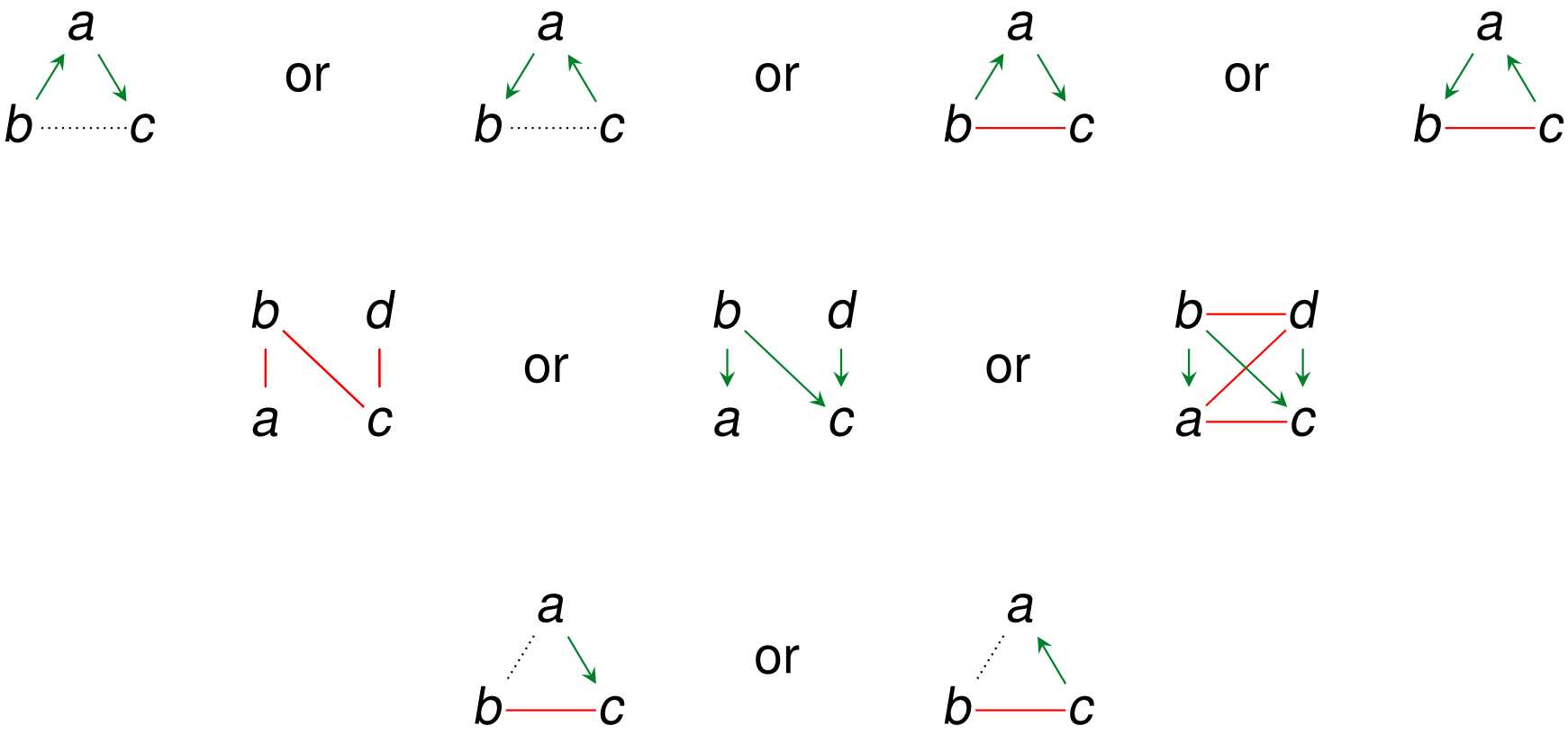
$[\![a]\!]$	$[\![a^\perp]\!]$	$[\![A \parr B]\!] = [\![A]\!] \parr [\![B]\!]$	$[\![A \triangleleft B]\!] = [\![A]\!] \triangleleft [\![B]\!]$	$[\![A \otimes B]\!] = [\![A]\!] \otimes [\![B]\!]$
a	$a^\perp$			

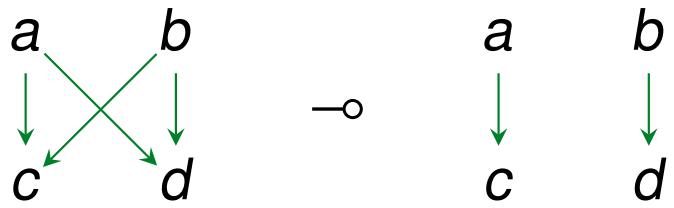


[2] Retoré '93, Guglielmi '07

Formulas and Graphs

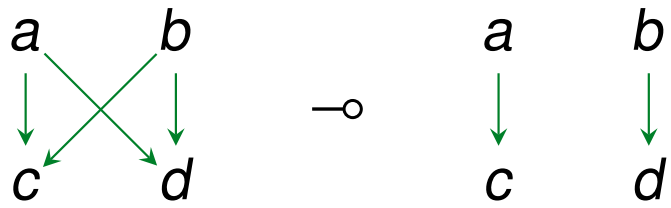
A graph containing an induced subgraph of the following shape cannot be represented by a formula:





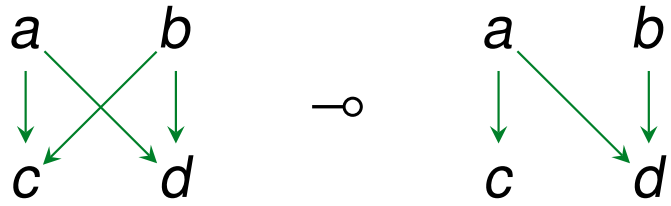
$$(a \parr b) \triangleleft (c \parr d) \multimap (a \triangleleft c) \parr (b \triangleleft d)$$

Provable in BV $\subset$ Pomset

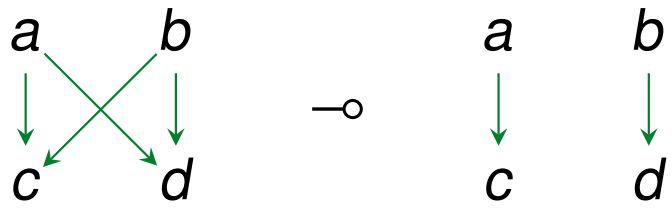


$(a \parr b) \triangleleft (c \parr d) \multimap (a \triangleleft c) \parr (b \triangleleft d)$

Provable in BV $\subset$ Pomset

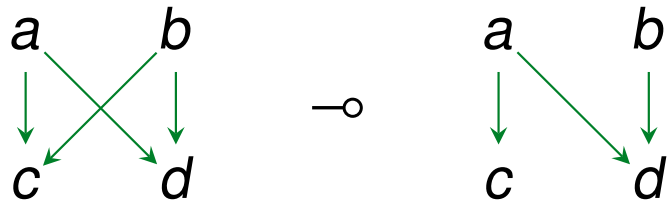


$(a \parr b) \triangleleft (c \parr d) \multimap$ NOT A FORMULA

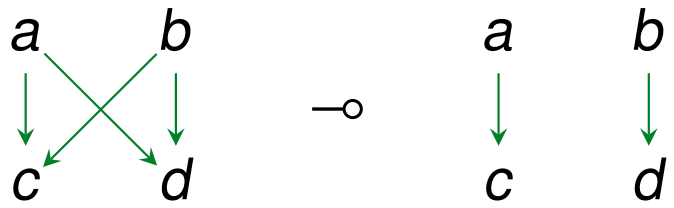


$(a \parr b) \triangleleft (c \parr d) \multimap (a \triangleleft c) \parr (b \triangleleft d)$

Provable in BV $\subset$ Pomset
(series-parallel orders)

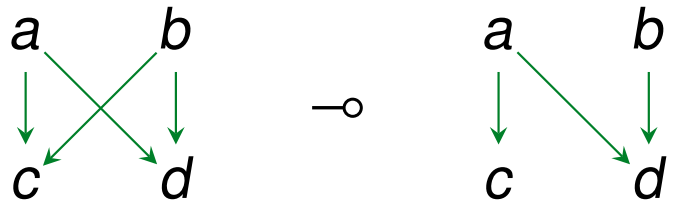


$(a \parr b) \triangleleft (c \parr d) \multimap$ NOT A FORMULA



$(a \parr b) \triangleleft (c \parr d) \multimap (a \triangleleft c) \parr (b \triangleleft d)$

Provable in BV $\subset$ Pomset
(series-parallel orders)



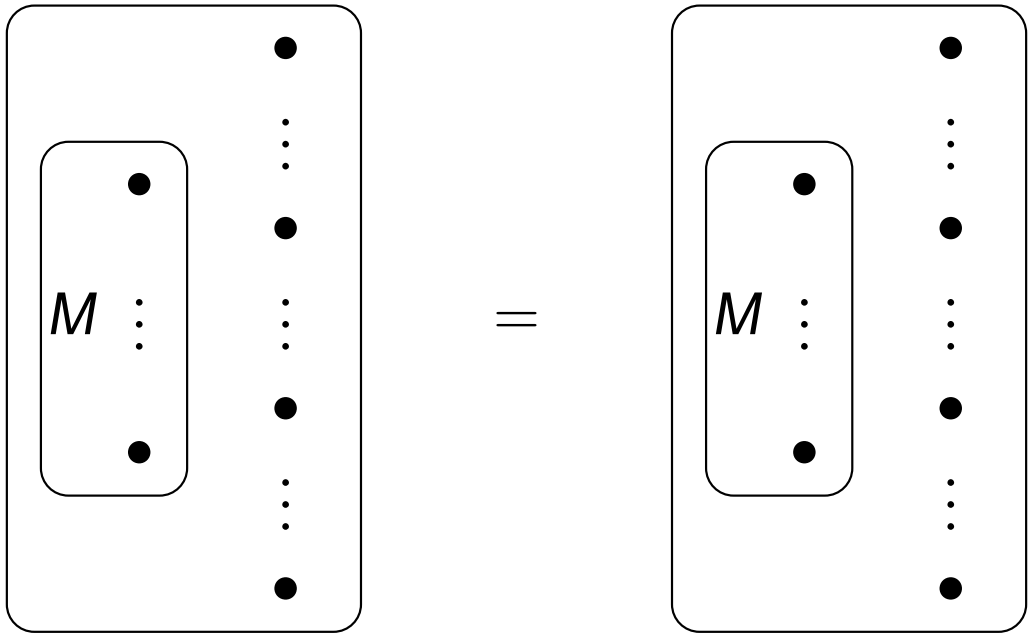
$(a \parr b) \triangleleft (c \parr d) \multimap$ NOT A FORMULA

Provable in $\mathrm{GV}^{\mathrm{sl}}$
(our result)

Preliminaries on Graphs

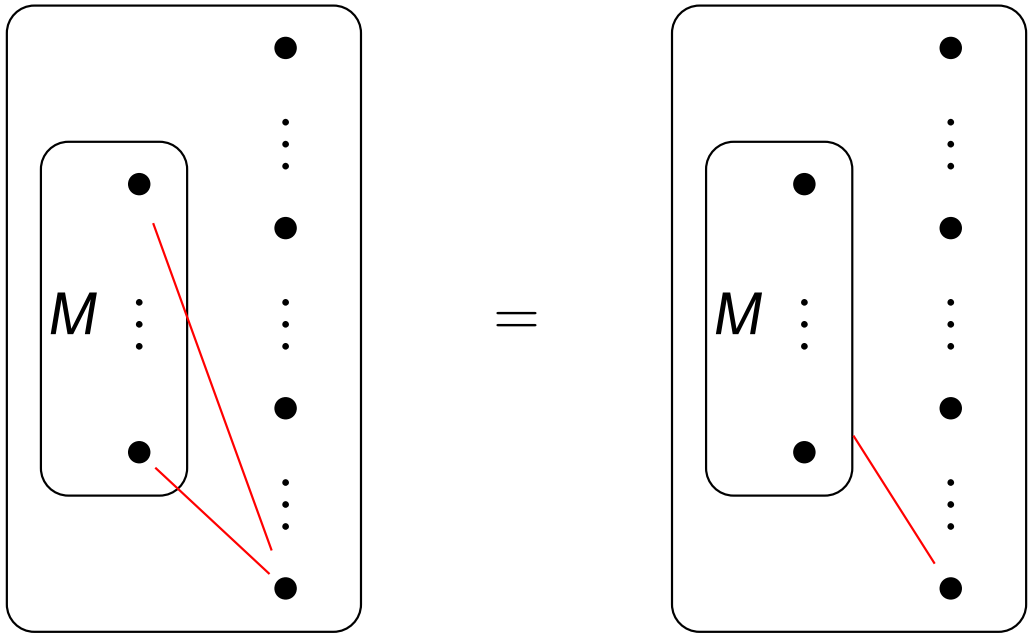
Modular Decomposition

A ***module*** of a graph $G = H[M]$ is a set of vertices M s.t.



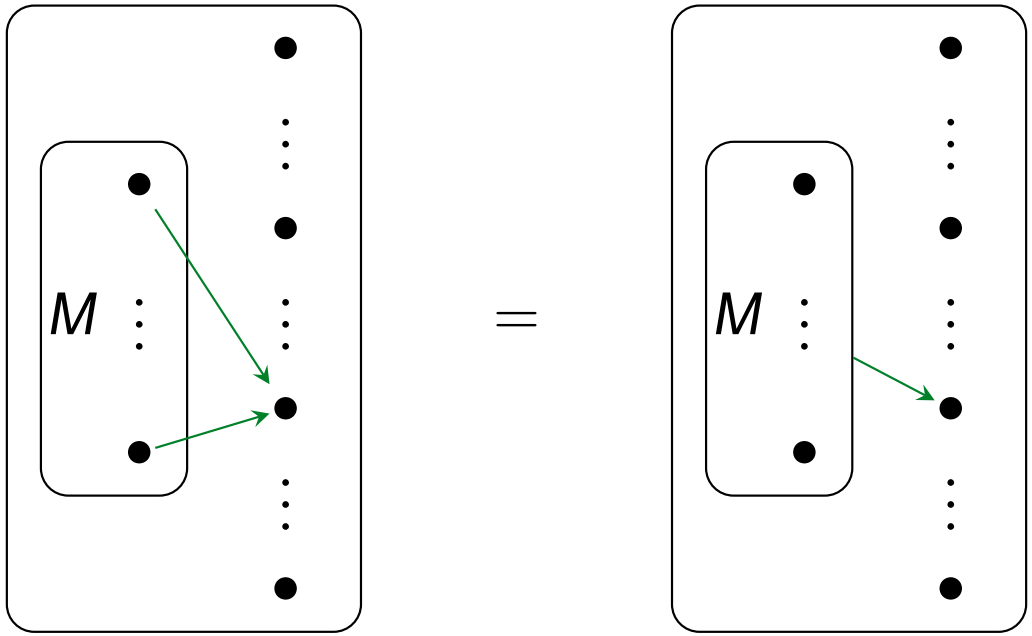
Modular Decomposition

A ***module*** of a graph $G = H[M]$ is a set of vertices M s.t.



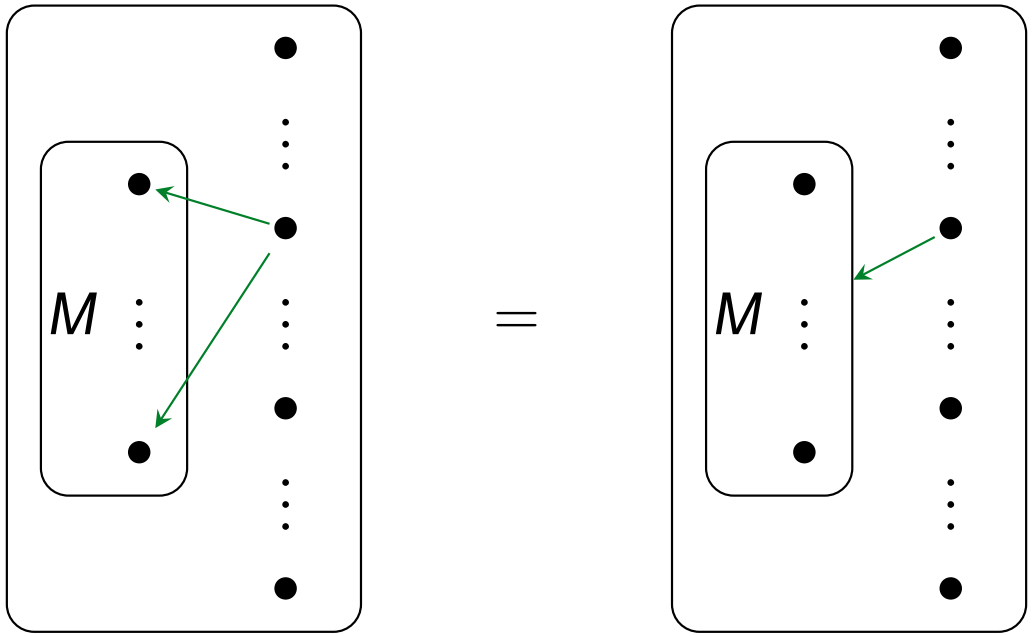
Modular Decomposition

A ***module*** of a graph $G = H[M]$ is a set of vertices M s.t.



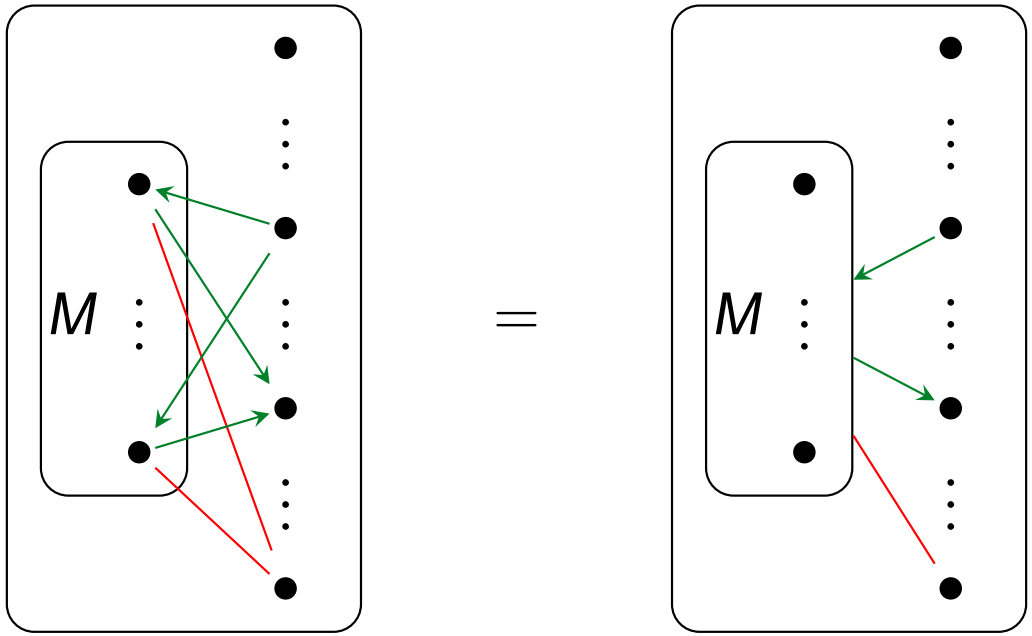
Modular Decomposition

A ***module*** of a graph $G = H[M]$ is a set of vertices M s.t.



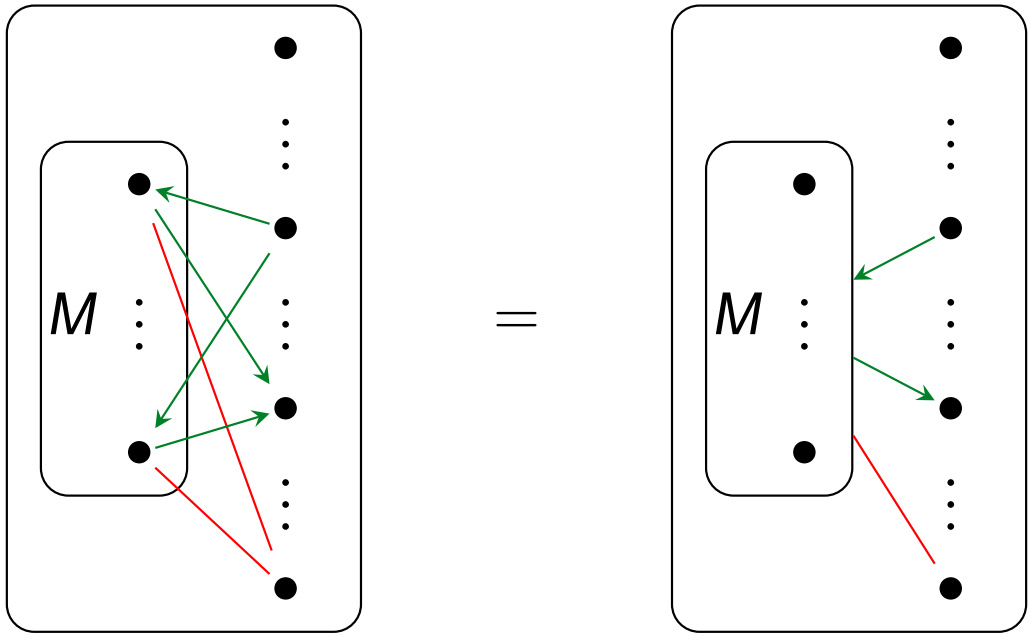
Modular Decomposition

A ***module*** of a graph $G = H[M]$ is a set of vertices M s.t.

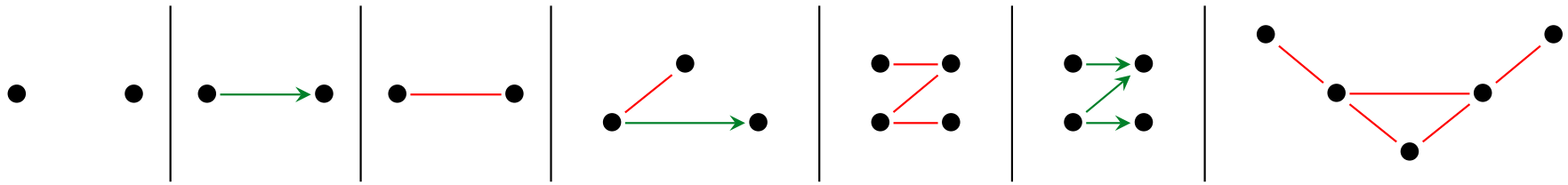


Modular Decomposition

A ***module*** of a graph $G = H[M]$ is a set of vertices M s.t.



A graph G is ***prime*** if it has modules V_G, $\emptyset$ and $\{x\}$ for all $x \in V_G$.



If G has n vertices and $H_1, \ldots, H_n$ graphs,
then we use G as a logic connective and we write $G(\!|H_1, \ldots, H_n|\!)$

$$\parr: \quad \bullet \quad \bullet \qquad \parr(\!|G, H|\!) = G \parr H$$

$$\triangleleft: \quad \bullet \longrightarrow \bullet \qquad \triangleleft(\!|G, H|\!) = G \triangleleft H$$

$$\otimes: \quad \bullet \text{---} \bullet \qquad \otimes(\!|G, H|\!) = G \otimes H$$

If G has n vertices and $H_1, \ldots, H_n$ graphs,
then we use G as a logic connective and we write $G(\!|H_1, \ldots, H_n|\!)$

$\parr$: $\parr(\!|G, H|\!) = G \parr H$ | $\triangleleft$: $\triangleleft(\!|G, H|\!) = G \triangleleft H$ | $\otimes$: $\otimes(\!|G, H|\!) = G \otimes H$

Lemma (Modular decomposition of graphs (Gallai '75))

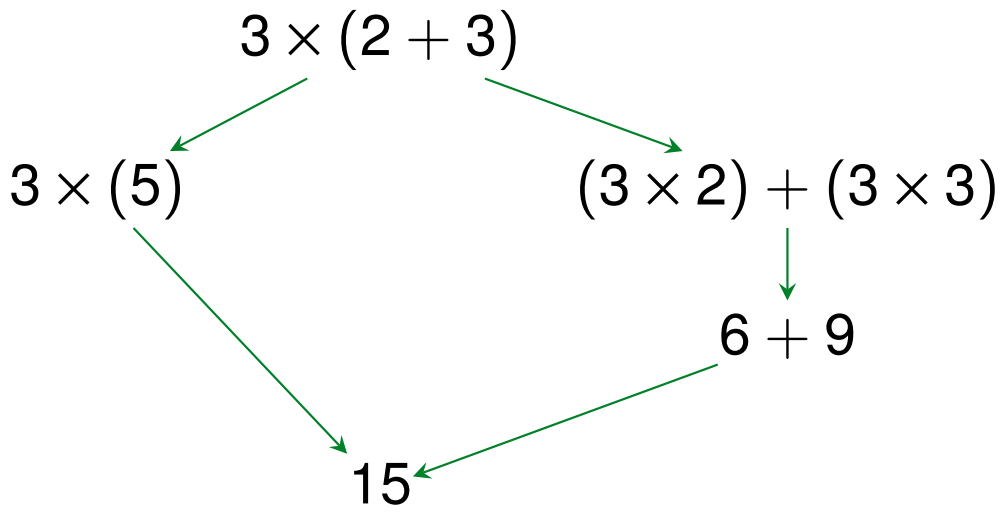
If $G \neq \varnothing$ is a graph, then we have exactly one of the following cases:

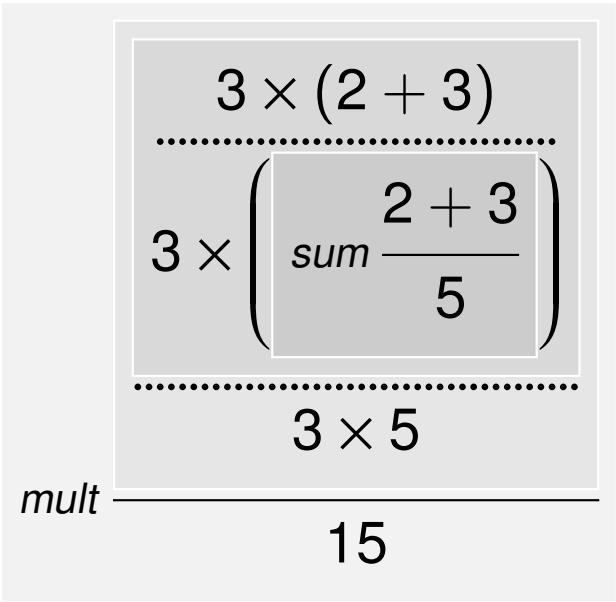
(i) *G is a singleton graph*

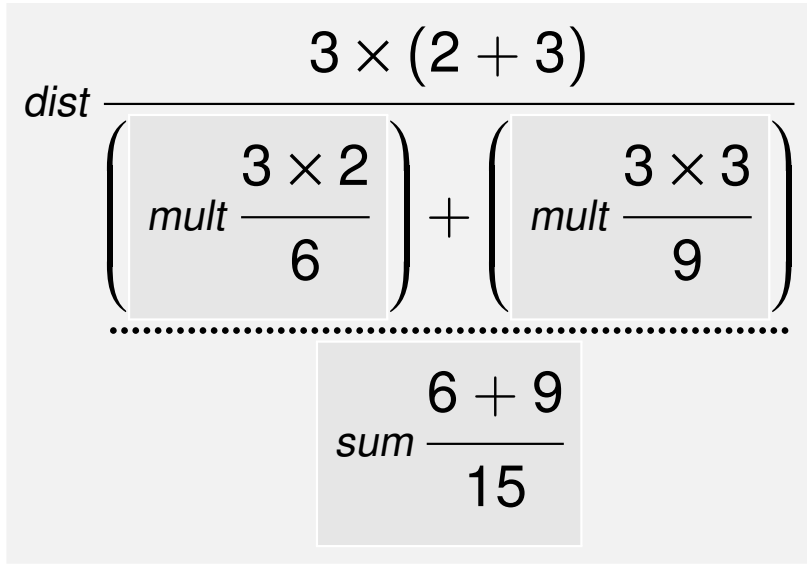
(ii) *$G = P(\!|A_1, \ldots, A_n|\!)$ for a prime graph P*

Graphical proof systems

On Deep Inference


3 × (2 + 3)
3 × (5)
(3 × 2) + (3 × 3)
6 + 9
15


3 × (2 + 3)
3 × (sum 2 + 3 / 5)
3 × 5
mult
15


dist
3 × (2 + 3)
(mult 3 × 2 / 6) + (mult 3 × 3 / 9)
sum 6 + 9 / 15

Deep inference does the same!

- $$\begin{array}{c} H_1 \\ \mathcal{D}_1 \parallel \\ G_1 \end{array} \quad \text{and} \quad \begin{array}{c} H_2 \\ \mathcal{D}_2 \parallel \\ G_2 \end{array} \quad \text{and} \quad \mathit{rule}\ \frac{G_1}{H_2} \quad \Longrightarrow \quad \mathit{rule}\ \frac{\begin{array}{c} H_1 \\ \mathcal{D}_1 \parallel \\ G_1 \end{array}}{\begin{array}{c} H_2 \\ \mathcal{D}_2 \parallel \\ G_2 \end{array}}$$

- $$\begin{array}{c} H_i \\ \mathcal{D}_i \parallel \\ G_i \end{array} \quad \text{and} \quad P \text{ an } n\text{-ary connective} \quad \Longrightarrow \quad P\left(\begin{array}{c} H_1 \\ \mathcal{D}_1 \parallel \\ G_1 \end{array}, \ldots, \begin{array}{c} H_n \\ \mathcal{D}_n \parallel \\ G_n \end{array} \right)$$

The rules

$$\mathrm{ai}{\downarrow}\ \frac{\varnothing}{a^{\perp} \parr a} \qquad \mathrm{ai}{\uparrow}\ \frac{a^{\perp} \otimes a}{\varnothing}$$

$$\mathrm{s}_{\parr}\ \frac{P(\!|M_1, \ldots, M_{i-1}, \mathbf{M_i} \parr N, M_{i+1}, \ldots M_n|\!)}{\mathbf{M_i} \parr P(\!|M_1, \ldots, M_{i-1}, N, M_{i+1}, \ \ldots, M_n|\!)} \qquad \mathrm{s}_{\otimes}\ \frac{\mathbf{M_i} \otimes P(\!|M_1, \ldots, M_{i-1}, N, M_{i+1}, \ldots, M_n|\!)}{P(\!|M_1, \ldots, M_{i-1}, \mathbf{M_i} \otimes N, M_{i+1}, \ldots, M_n|\!)}$$

$$\mathrm{p}{\downarrow}\ \frac{(\mathbf{M_1} \parr N_1) \otimes \cdots \otimes (\mathbf{M_n} \parr N_n)}{R^{\perp}(\!|\mathbf{M_1}, \ldots, \mathbf{M_n}|\!) \parr R(\!|N_1, \ldots, N_n|\!)} \qquad \mathrm{p}{\uparrow}\ \frac{R(\!|\mathbf{M_1}, \ldots, \mathbf{M_n}|\!) \otimes R^{\perp}(\!|N_1, \ldots, N_n|\!)}{(\mathbf{M_1} \otimes N_1) \parr \cdots \parr (\mathbf{M_n} \otimes N_n)}$$

$$\mathrm{q}{\downarrow}\ \frac{Q(\!|\mathbf{M_1} \parr N_1, \ldots, \mathbf{M_n} \parr N_n|\!)}{Q^{\perp}(\!|\mathbf{M_1}, \ldots, \mathbf{M_n}|\!) \parr Q(\!|N_1, \ldots, N_n|\!)} \qquad \mathrm{q}{\uparrow}\ \frac{Q(\!|\mathbf{M_1}, \ldots, \mathbf{M_n}|\!) \otimes Q^{\perp}(\!|N_1, \ldots, N_n|\!)}{Q(\!|\mathbf{M_1} \otimes N_1, \ldots, \mathbf{M_n} \otimes N_n|\!)}$$

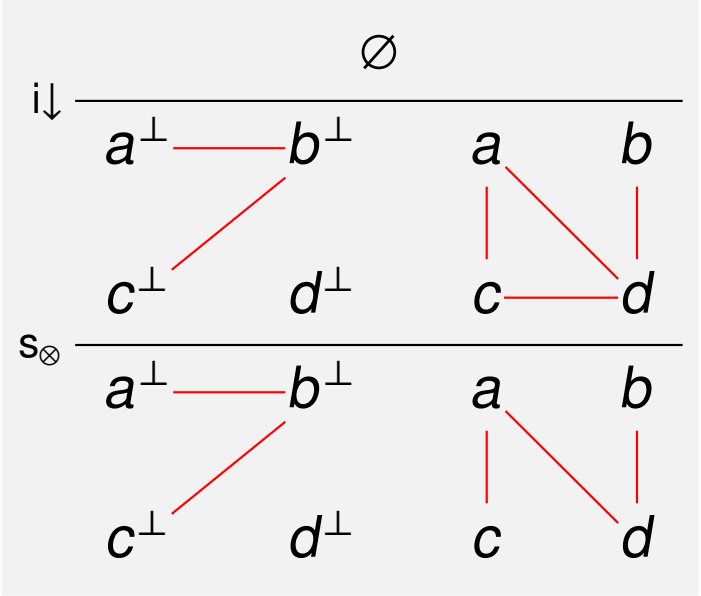
$$\mathrm{qm}\ \frac{Q(\!|L_1 \parr J_1, \ldots, L_n \parr J_n|\!)}{Q(\!|L_1, \ldots, L_n|\!) \parr Q(\!|J_1, \ldots, J_n|\!)}$$

$$\mathrm{sl}\ \frac{Q(\!|M_1, \ldots, M_k, \varnothing, \ldots, \varnothing|\!) \triangleleft Q(\!|\varnothing, \ldots, \varnothing, M_{k+1}, \ldots, M_n|\!)}{Q(\!|M_1, \ldots, M_n|\!)}$$

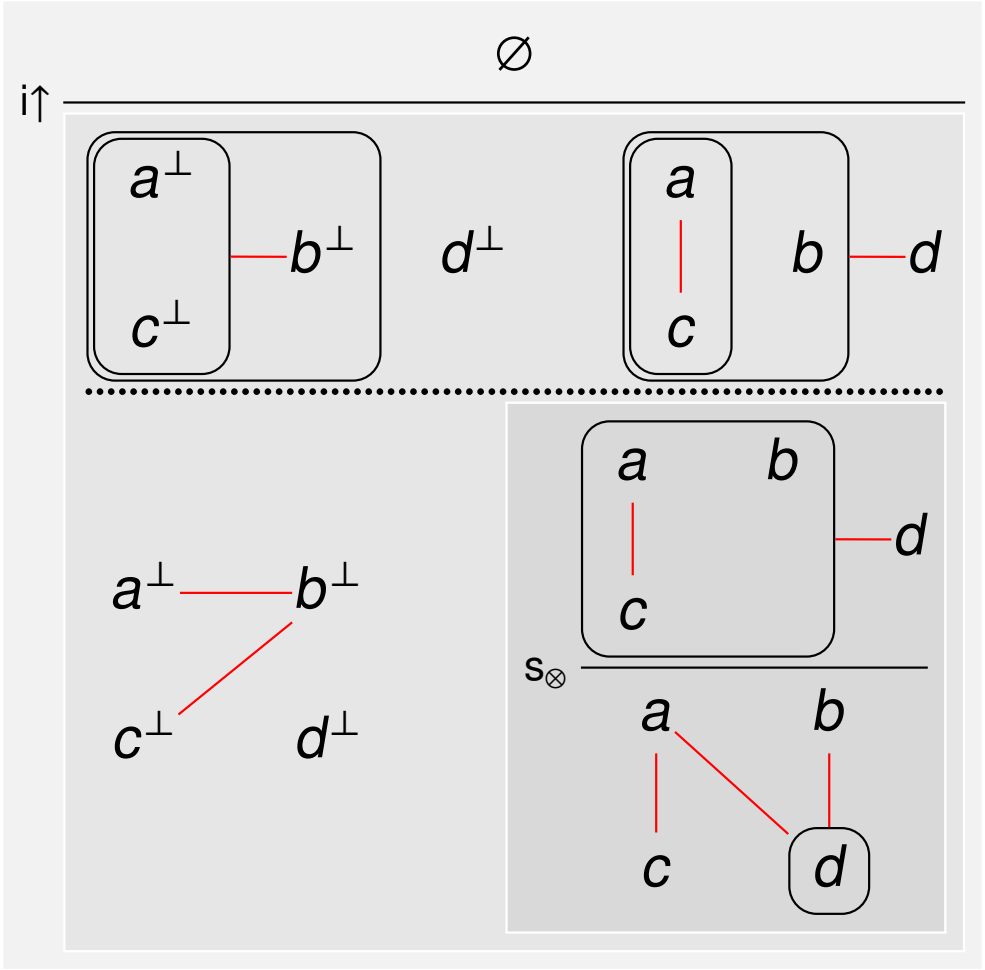
P, Q and R prime graphs with R red-white and Q green-white. M_i and $L_i \parr J_i$ are non-empty

Note: in this work graphs without three-color prime graphs in the modular decomposition

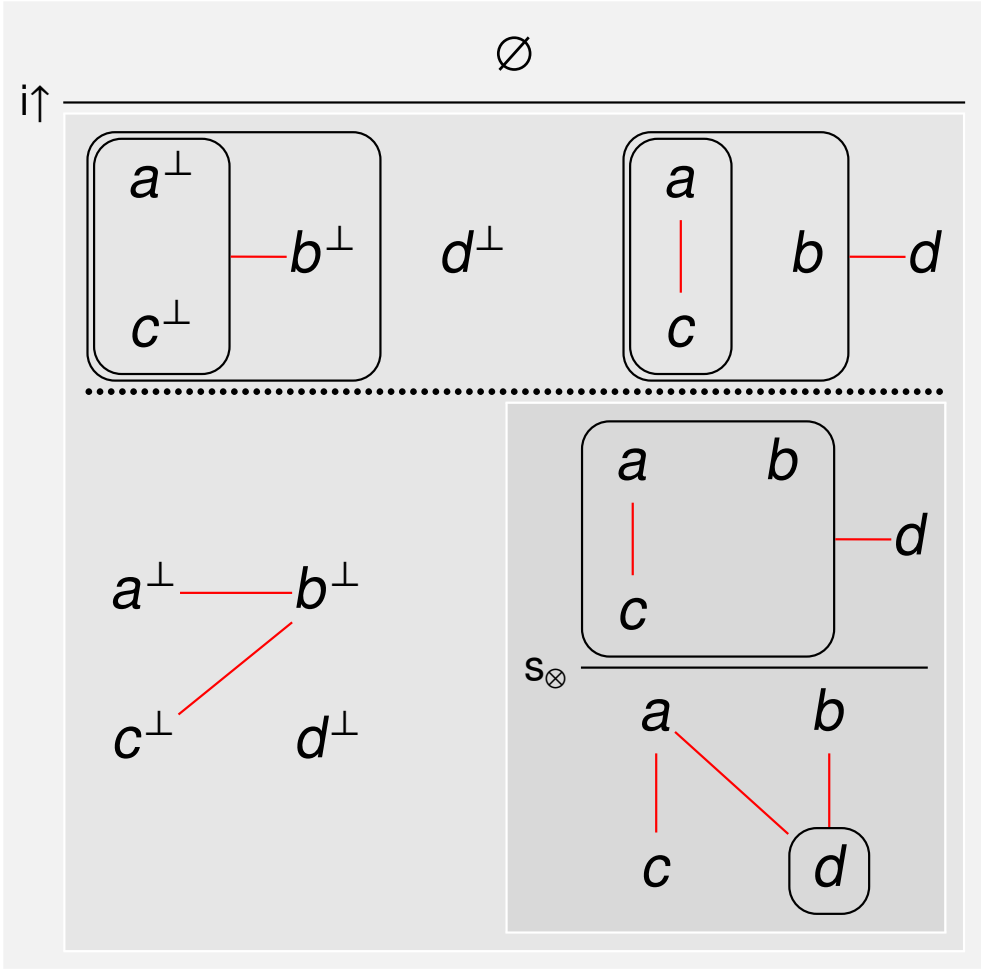
An example:


∅
i↓
$a^\perp$ $b^\perp$ a b
$c^\perp$ $d^\perp$ c d
$s_\otimes$
$a^\perp$ $b^\perp$ a b
$c^\perp$ $d^\perp$ c d

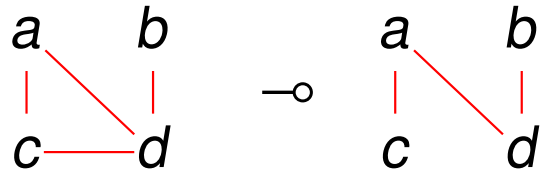
An example:

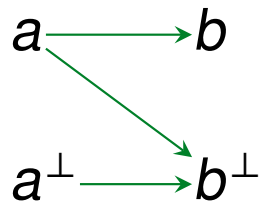

∅
i↑
$a^\perp$
$b^\perp$
$c^\perp$
$d^\perp$
a
b
c
d
$a^\perp$
$b^\perp$
$c^\perp$
$d^\perp$
a
b
c
d
$s_\otimes$
a
b
c
d

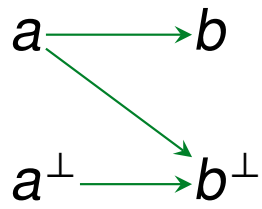
An example:



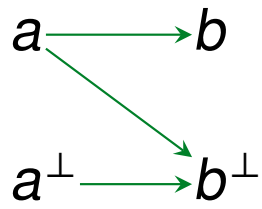
That is: $\vdash_{GS}$



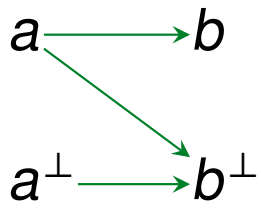

a
b
a⊥
b⊥



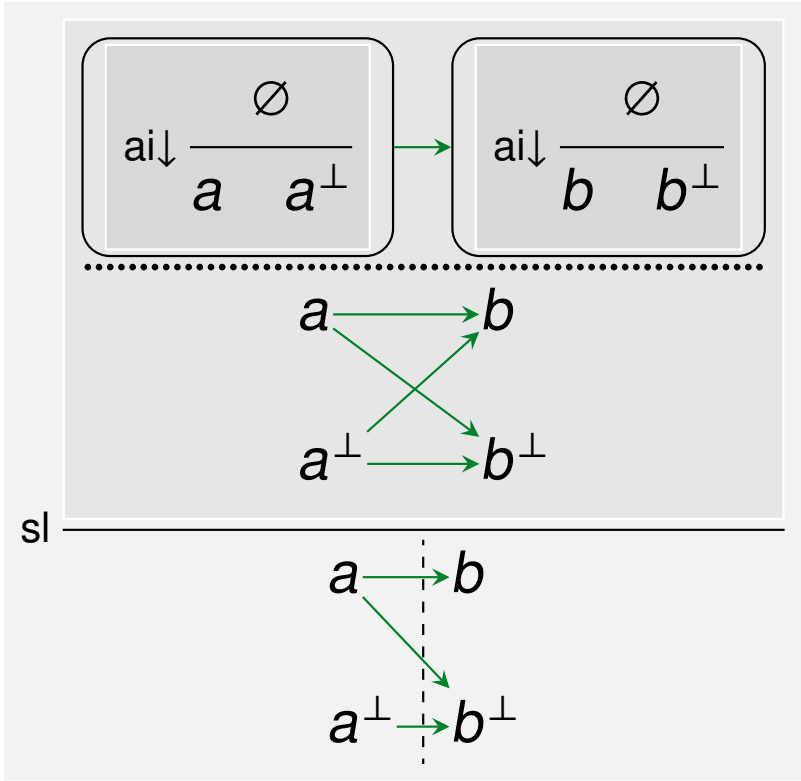
The rule sl refines a partial order *slicing* a “before” and an “after”

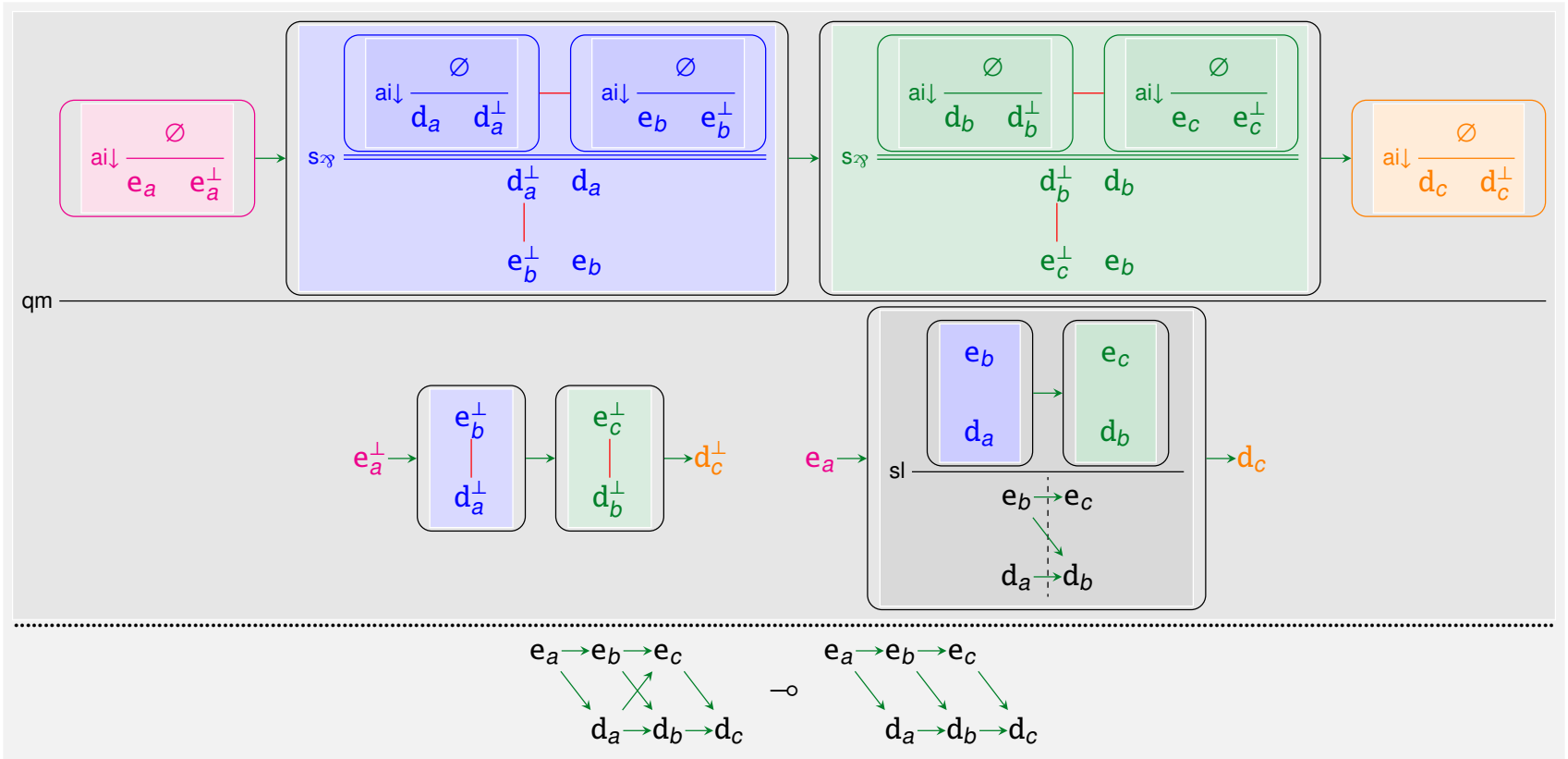


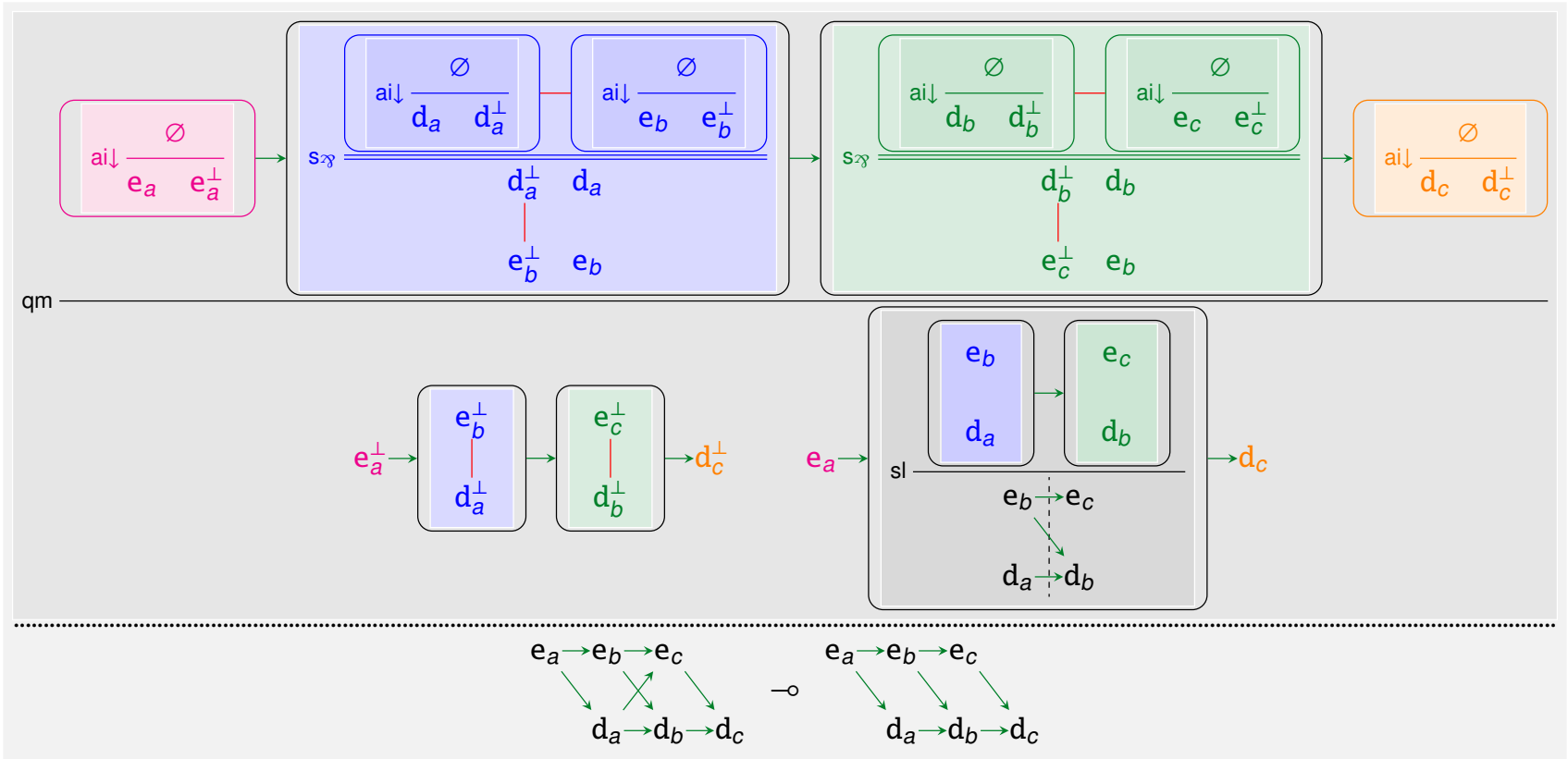
The rule sl refines a partial order *slicing* a “before” and an “after”
This rules implements Logical Clocks!



The rule sl refines a partial order *slicing* a “before” and an “after”
This rules implements Logical Clocks!




ai↓ ∅ / e_a $e_a^\perp$
$s_\parr$ ai↓ ∅ / d_a $d_a^\perp$ ai↓ ∅ / e_b $e_b^\perp$ — $d_a^\perp$ d_a — $e_b^\perp$ e_b
$s_\parr$ ai↓ ∅ / d_b $d_b^\perp$ ai↓ ∅ / e_c $e_c^\perp$ — $d_b^\perp$ d_b — $e_c^\perp$ e_b
ai↓ ∅ / d_c $d_c^\perp$
qm
$e_a^\perp$ $e_b^\perp$ $d_a^\perp$ $e_c^\perp$ $d_b^\perp$ $d_c^\perp$
e_a e_b d_a e_c d_b sl d_c
$e_b \to e_c$
$d_a \to d_b$
$e_a \to e_b \to e_c$
$d_a \to d_b \to d_c$
$\multimap$
$e_a \to e_b \to e_c$
$d_a \to d_b \to d_c$



We can proof theoretically prove that 2-Queues can simulate 3-Queues

Transitivity of $\multimap$

Remark

if $A \multimap B$ and $B \multimap C$, then $A \multimap C$

Remark

if $(A^{\perp} \parr B)$ and $(B^{\perp} \parr C)$, then $(A^{\perp} \parr C)$

Remark

if $(A^{\perp} \parr B) \otimes (B^{\perp} \parr C)$, then $(A^{\perp} \parr C)$

Remark

if $(A^{\perp} \parr B) \otimes (B^{\perp} \parr C)$, then $(A^{\perp} \parr C)$

$$\mathsf{p}\downarrow \frac{\begin{array}{c}\varnothing \\ \| \\ A^{\perp} \parr B\end{array} \otimes \begin{array}{c}\varnothing \\ \| \\ B^{\perp} \parr C\end{array}}{= \dfrac{A^{\perp} \parr \mathsf{i}\uparrow \dfrac{B^{\perp} \otimes B}{\varnothing} \parr C}{A^{\perp} \parr C}}$$

Remark

if $(A^{\perp} \parr B) \otimes (B^{\perp} \parr C)$, then $(A^{\perp} \parr C)$

$$\mathrm{p}\downarrow \frac{\dfrac{\varnothing}{\|}\ A^{\perp} \parr B \quad \otimes \quad \dfrac{\varnothing}{\|}\ B^{\perp} \parr C}{= \dfrac{A^{\perp} \parr \mathrm{i}\uparrow \dfrac{B^{\perp} \otimes B}{\varnothing} \parr C}{A^{\perp} \parr C}}$$

Cut Elimination = the rule $\mathrm{i}\uparrow \dfrac{A \otimes A^{\perp}}{\varnothing}$ is admissible

Theorem

The rule $\mathsf{i}\uparrow$ *is derivable in* $\{\mathsf{ai}\uparrow, \mathsf{p}\uparrow, \mathsf{q}\uparrow\}$

Theorem

- *rules* $\{\mathsf{ai}\uparrow, \mathsf{p}\uparrow, \mathsf{s}_{\otimes}\}$ *are admissible in* $\mathsf{GS} = \{\mathsf{ai}\downarrow, \mathsf{s}_{\parr}, \mathsf{p}\downarrow\}$
- *rules* $\{\mathsf{ai}\uparrow, \mathsf{p}\uparrow, \mathsf{q}\uparrow\}$ *are admissible in* $\mathsf{GV} = \{\mathsf{ai}\downarrow, \mathsf{s}_{\parr}, \mathsf{s}_{\otimes}, \mathsf{p}\downarrow, \mathsf{q}\downarrow, \mathsf{qm}\}$
- *rules* $\{\mathsf{ai}\uparrow, \mathsf{p}\uparrow, \mathsf{q}\uparrow\}$ *are admissible in* $\mathsf{GV}^{\mathsf{sl}} = \{\mathsf{ai}\downarrow, \mathsf{s}_{\parr}, \mathsf{s}_{\otimes}, \mathsf{p}\downarrow, \mathsf{q}\downarrow, \mathsf{qm}, \mathsf{sl}\}$

$$\mathrm{ai}{\downarrow}\ \frac{\varnothing}{a^{\perp} \parr a} \qquad \mathrm{ai}{\uparrow}\ \frac{a^{\perp} \otimes a}{\varnothing}$$

$$\mathrm{s}_{\parr}\ \frac{P(\!|M_1, \ldots, M_{i-1}, \mathbf{M_i} \parr N, M_{i+1}, \ldots M_n|\!)}{\mathbf{M_i} \parr P(\!|M_1, \ldots, M_{i-1}, N, M_{i+1}, \ \ldots, M_n|\!)} \qquad \mathrm{s}_{\otimes}\ \frac{\mathbf{M_i} \otimes P(\!|M_1, \ldots, M_{i-1}, N, M_{i+1}, \ldots, M_n|\!)}{P(\!|M_1, \ldots, M_{i-1}, \mathbf{M_i} \otimes N, M_{i+1}, \ldots, M_n|\!)}$$

$$\mathrm{p}{\downarrow}\ \frac{(\mathbf{M_1} \parr N_1) \otimes \cdots \otimes (\mathbf{M_n} \parr N_n)}{R^{\perp}(\!|\mathbf{M_1}, \ldots, \mathbf{M_n}|\!) \parr R(\!|N_1, \ldots, N_n|\!)} \qquad \mathrm{p}{\uparrow}\ \frac{R(\!|\mathbf{M_1}, \ldots, \mathbf{M_n}|\!) \otimes R^{\perp}(\!|N_1, \ldots, N_n|\!)}{(\mathbf{M_1} \otimes N_1) \parr \cdots \parr (\mathbf{M_n} \otimes N_n)}$$

$$\mathrm{q}{\downarrow}\ \frac{Q(\!|\mathbf{M_1} \parr N_1, \ldots, \mathbf{M_n} \parr N_n|\!)}{Q^{\perp}(\!|\mathbf{M_1}, \ldots, \mathbf{M_n}|\!) \parr Q(\!|N_1, \ldots, N_n|\!)} \qquad \mathrm{q}{\uparrow}\ \frac{Q(\!|\mathbf{M_1}, \ldots, \mathbf{M_n}|\!) \otimes Q^{\perp}(\!|N_1, \ldots, N_n|\!)}{Q(\!|\mathbf{M_1} \otimes N_1, \ldots, \mathbf{M_n} \otimes N_n|\!)}$$

$$\mathrm{qm}\ \frac{Q(\!|L_1 \parr J_1, \ldots, L_n \parr J_n|\!)}{Q(\!|L_1, \ldots, L_n|\!) \parr Q(\!|J_1, \ldots, J_n|\!)}$$

$$\mathrm{sl}\ \frac{Q(\!|M_1, \ldots, M_k, \varnothing, \ldots, \varnothing|\!) \triangleleft Q(\!|\varnothing, \ldots, \varnothing, M_{k+1}, \ldots, M_n|\!)}{Q(\!|M_1, \ldots, M_n|\!)}$$

How to prove “cut-elimination”?

How to prove “cut-elimination”?

Splitting:

Pick a connected component of a provable $G \parr P(\!|M_1, \ldots, M_n|\!)$.
You can apply rules to G until you have a rule destroying P.

How to prove “cut-elimination”?

Splitting:

Pick a connected component of a provable $G \parr P(\!|M_1, \ldots, M_n|\!)$.
You can apply rules to G until you have a rule destroying P.

Context-Reduction:

You can apply splitting to a connector P deep in a context.

How to prove “cut-elimination”?

Splitting:

Pick a connected component of a provable $G \parr P(\!|M_1, \ldots, M_n|\!)$.
You can apply rules to G until you have a rule destroying P.

Context-Reduction:

You can apply splitting to a connector P deep in a context.

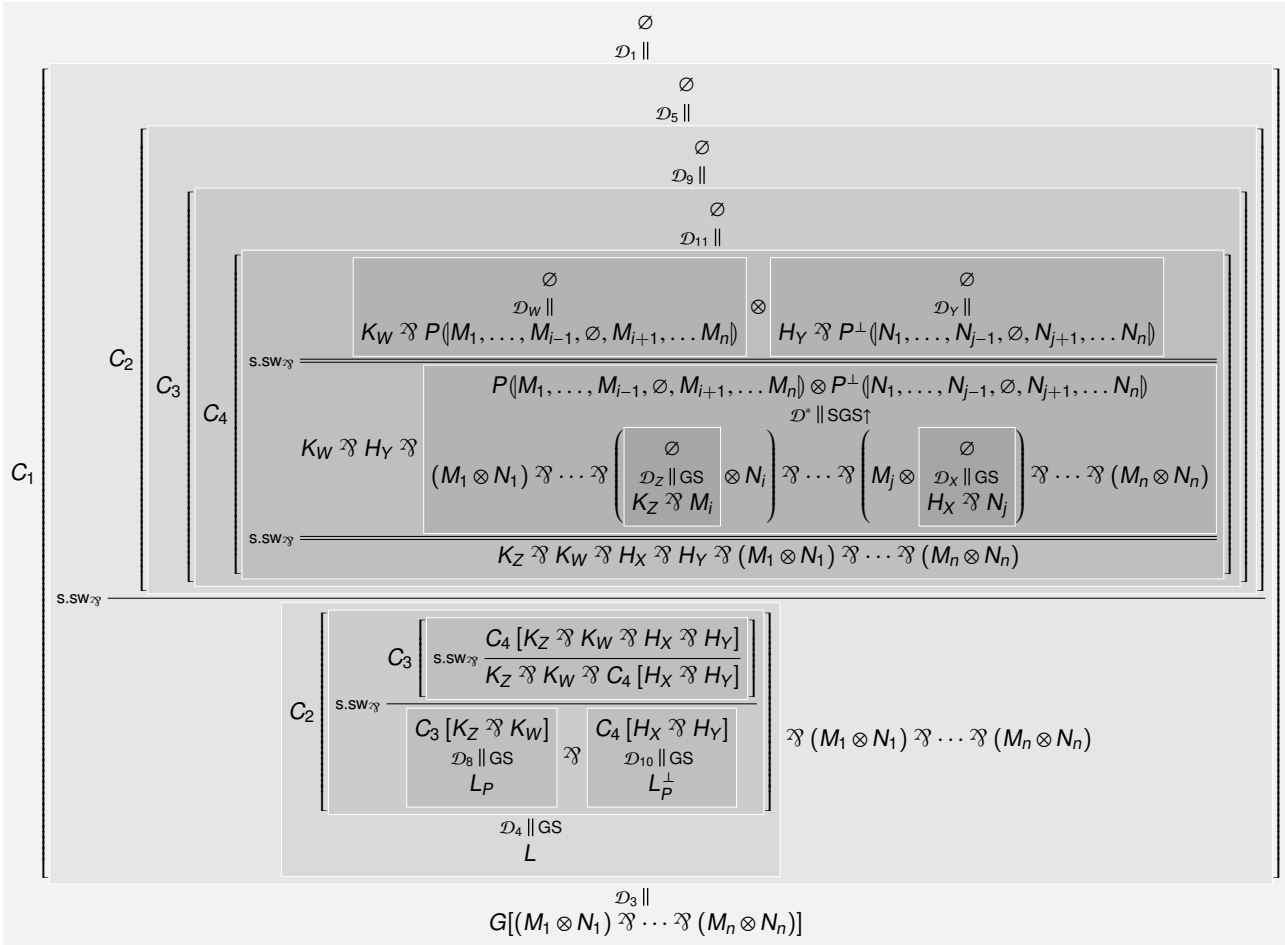
Up-elimination:

If the premise of an up-rule is provable, then its conclusion also is.

Remark: up-rules elimination is not trivial

Remark: up-rules elimination is not trivial

One case of the p↑-elimination in GS:



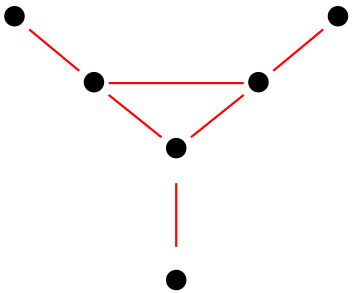
Conservativity

$$\vdash_{\mathsf{GX}} [\![F]\!] \implies \vdash_{\mathsf{X}} F$$

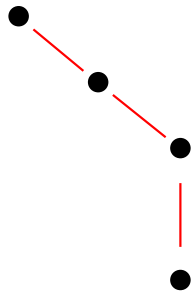
Note: $A \star \varnothing = \star(A, \varnothing) = A$

Note: $A \star \varnothing = \star(A, \varnothing) = A$

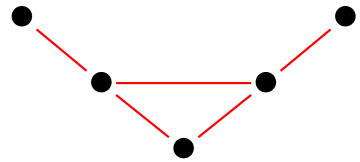
A new notion of *"sub-formula"* analyticity arises from this work:



has sub-connective (e.g.)

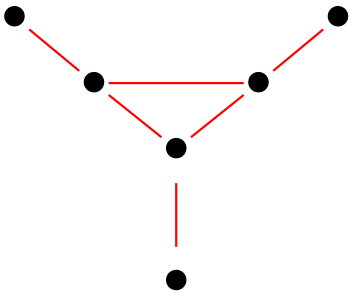


and

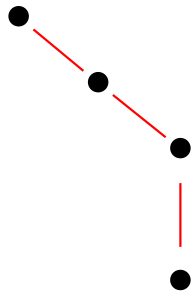


Note: $A \star \varnothing = \star(A, \varnothing) = A$

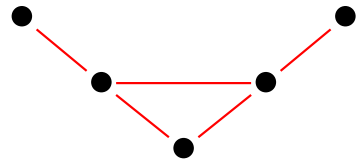
A new notion of *"sub-formula"* analyticity arises from this work:



has sub-connective (e.g.)

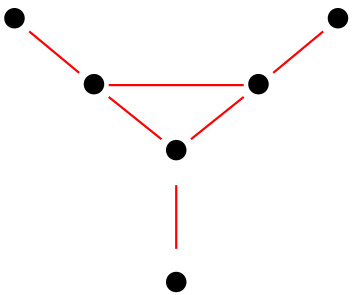


and

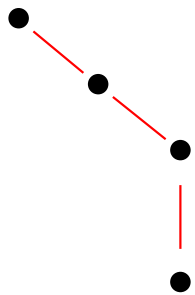


Proper sub-connective: sub-connective with $\frown \neq \varnothing$

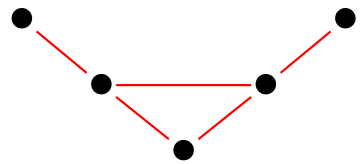
Note: $A \star \varnothing = \star(A, \varnothing) = A$
A new notion of *"sub-formula"* analyticity arises from this work:



has sub-connective (e.g.)

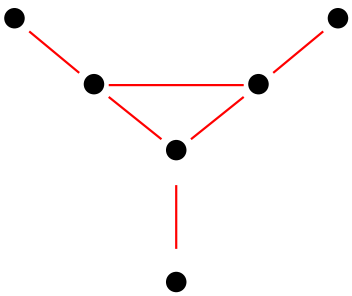
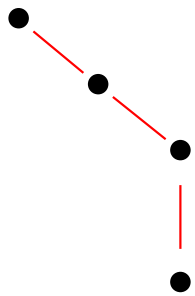
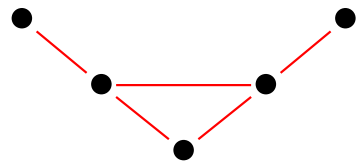


and



Proper sub-connective: sub-connective with $\frown \neq \varnothing$
Analytic proof: only proper sub-connective of the conclusion

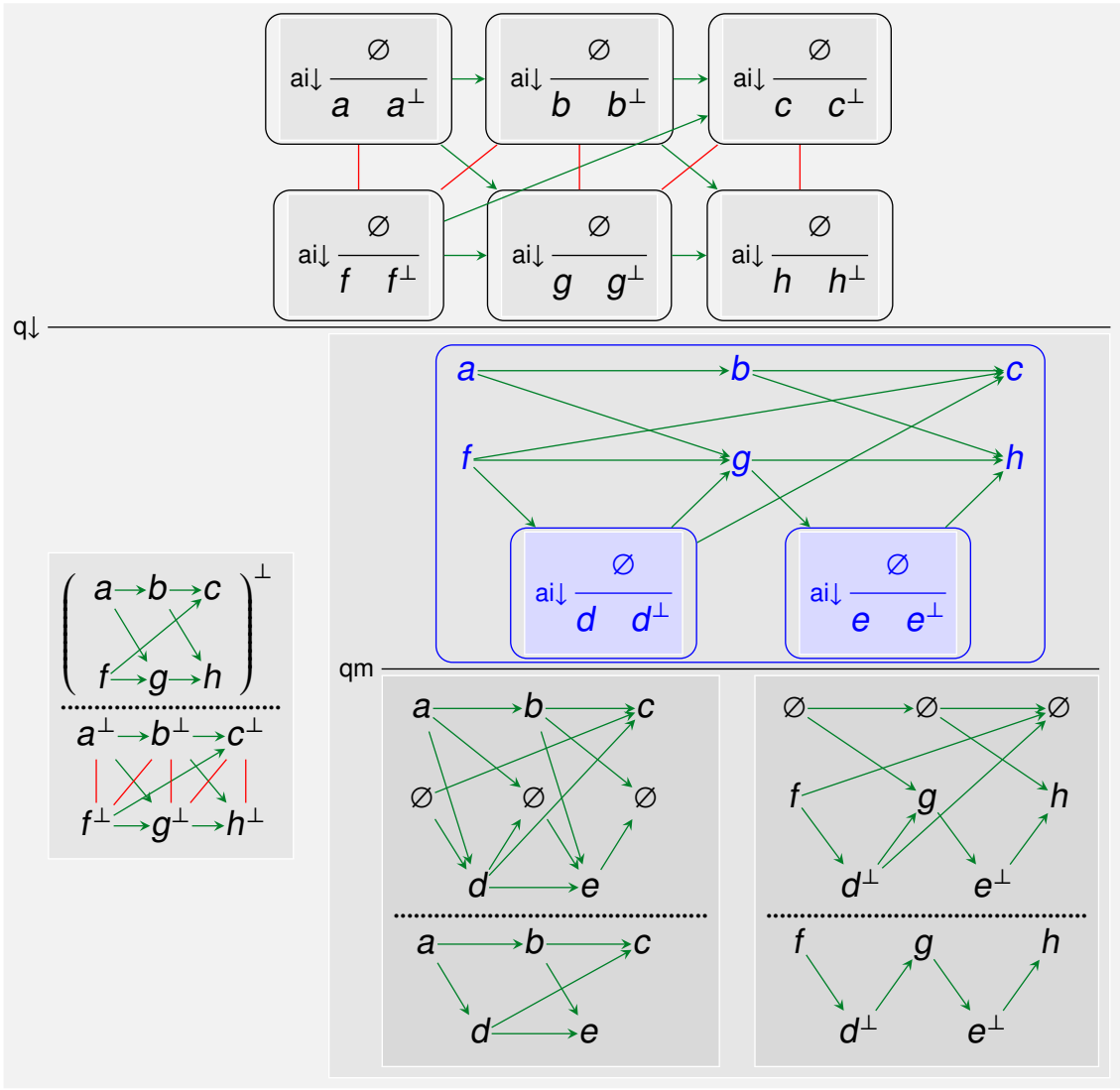
Note: $A \star \varnothing = \star(A, \varnothing) = A$
A new notion of *"sub-formula"* analyticity arises from this work:


has sub-connective (e.g.)

and


Proper sub-connective: sub-connective with $\frown \neq \varnothing$
Analytic proof: only proper sub-connective of the conclusion

Theorem

In GV *and* $\mathrm{GV}^{\mathrm{sl}}$*, if G is provable, than G admits an analytic proof.*


ai↓ ∅ / a a⊥
ai↓ ∅ / b b⊥
ai↓ ∅ / c c⊥
ai↓ ∅ / f f⊥
ai↓ ∅ / g g⊥
ai↓ ∅ / h h⊥
q↓
a b c
f g h
ai↓ ∅ / d d⊥
ai↓ ∅ / e e⊥
qm
a b c
∅ ∅ ∅
d e
a b c
d e
∅ ∅ ∅
f g h
d⊥ e⊥
f g h
d⊥ e⊥
a⊥ b⊥ c⊥
f⊥ g⊥ h⊥

Conclusions and Future Works

Our results:

(-) GS, GV and GV^{sl} are proof systems [in the sense of Cook-Reckhow]

Our results:

(-) GS, GV and GV^{sl} are proof systems [in the sense of Cook-Reckhow]

(-) Size of a proof is polynomialy bounded

(-) GS is a conservative extension of MLL

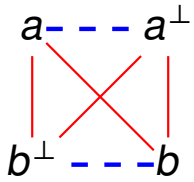
(-) GV and GV^{sl} are both conservative extensions of both BV and GS

Future works:

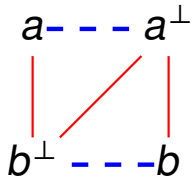
(-) Categorical and Algebraic Semantics

(-) Topological correctness criteria:

- Correctness criterion for GS



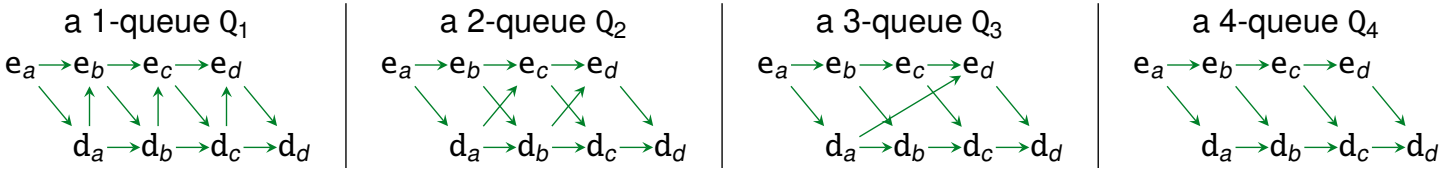
Satisfies Retoré's Criterion
Provable in GS



Satisfies Retoré's Criterion
NOT provable in GS

- Correctness criterion for BV $\Longrightarrow$ criteria for GV and GS

(-) Full homomorphism requires new tools



$\vdash_{BV} Q_1 \multimap Q_2 \quad \vdash_{GV^{sl}} Q_2 \multimap Q_3 \quad \vdash_{???} Q_3 \multimap Q_4$

(-) Applications to Games/Petri Nets/Event Structures/Concurrency/Verification

Thank you

Thank you

Questions?
Comments?
Feedbacks?